

Inductive Concept Retrieval and Query Answering with Semantic Knowledge Bases through Kernel Methods

Nicola Fanizzi and Claudia d'Amato

Dipartimento di Informatica, Università degli Studi di Bari
Campus Universitario, Via Orabona 4, 70125 Bari, Italy
{fanizzi|claudia.damato}@di.uniba.it

Abstract. This work deals with the application of kernel methods to structured relational settings such as semantic knowledge bases expressed in Description Logics. Our method integrates a novel kernel function for the \mathcal{ALC} logic in a *support vector machine* that could be set up to work with these representations. In particular, we present experiments where our method is applied to the tasks of concept retrieval and query answering on existing ontologies.

Key words: Inductive Concept Retrieval, Query Answering, Kernel Methods, Kernel Function, Description Logics, Semantic Web

1 Learning in Multi-Relational Settings

Many application domains, spanning from computational biology and chemistry to natural language processing, require operating on structured data representations. A new emerging domain is represented by the Semantic Web (SW) [1] where knowledge intensive manipulations on complex relational descriptions are foreseen to be performed by machines. In this context, *Description Logics* (DLs) [2] have been adopted as the core technology for ontology languages, such as OWL. This family of languages is endowed with well-founded semantics and reasoning services (see Sect. 2). Unfortunately, machine learning through logic-based methods is inherently intractable in multi-relational settings, unless language bias is imposed to constrain the representation. Yet, for the sake of tractability, only very simple DL languages have been considered so far.

Kernel methods [3] are a family of efficient statistical learning algorithms, including the *support vector machines* (SVMs), that have been effectively applied to a variety of tasks, recently also in domains that typically require structured representations [4, 5]. They can be very efficient because they map, by means of a kernel function, the original feature space of the considered data set into a high-dimensional space, where the learning task is simplified. However, such a mapping is not explicitly performed (*kernel trick*): it requires a sound definition of a positive definite kernel function on the feature space; the validity of such a function ensures that the embedding into a new space exists, so that it corresponds to the inner product in this space [3].

In this work, we exploit a kernel function for DLs representations, specifically for the \mathcal{ALC} logic [6]. It encodes a notion of similarity of individuals in this representation, based on both structural and semantic aspects of the reference representation (see

Sect. 3). By means of the resulting SVM, many tasks based on inductive classification can be tackled. Particularly, we demonstrate how to perform important inferences on semantic knowledge bases, namely concept retrieval and query answering. These tasks are generally grounded on merely deductive procedures which easily fail in case of (partially) inconsistent or incomplete knowledge. We show how the methods performs comparably well w.r.t. a standard deductive reasoner, allowing the suggestion of new knowledge that was not previously logically derivable. Indeed, the method was implemented and experimentally tested on artificial and real ontologies drawn from standard repositories as illustrated in Sect. 5.

2 Reference Representation Space

We will recall the basics of \mathcal{ALC} (see [2] for a thorough reference). Such a logic is not trivial as it is endowed with the basic constructors employed by the standard ontology languages and deductive reasoning is quite computationally expensive [7].

Descriptions are inductively defined starting with a set N_C of *primitive concept* names and a set N_R of *primitive roles*. Complex descriptions are built using primitive concepts and roles and the language constructors. The semantics of the descriptions is defined by an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set, the *domain* of the interpretation, and $\cdot^{\mathcal{I}}$ is the *interpretation function* that maps each $A \in N_C$ to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and each $R \in N_R$ to $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The *top* concept \top is interpreted as the whole domain $\Delta^{\mathcal{I}}$, while the *bottom* concept \perp corresponds to \emptyset . Complex descriptions can be built in \mathcal{ALC} using the following constructors. *Full negation*: given any description C , it is denoted $\neg C$ and amounts to $\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$. *Concept conjunction*, denoted by $C_1 \sqcap C_2$, yields an extension $C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$ and, dually, *concept disjunction*, denoted $C_1 \sqcup C_2$, yields the union $C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$. Finally, the *existential restriction*, denoted $\exists R.C$, is interpreted as the set $\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}((x, y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}})\}$ and the *value restriction* $\forall R.C$, has the extension $\{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}}((x, y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}})\}$.

The main inference is *subsumption* between concepts based on their semantics: given two descriptions C and D , C *subsumes* D , denoted by $C \sqsupseteq D$, iff for every interpretation \mathcal{I} it holds that $C^{\mathcal{I}} \supseteq D^{\mathcal{I}}$. When $C \sqsupseteq D$ and $D \sqsupseteq C$ then they are equivalent, denoted with $C \equiv D$.

A *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ contains a *TBox* \mathcal{T} and an *ABox* \mathcal{A} . \mathcal{T} is the set of definitions $C \equiv D$, meaning $C^{\mathcal{I}} = D^{\mathcal{I}}$, where C is the concept name and D is its description. \mathcal{A} contains assertions on the world state, e.g. $C(a)$ and $R(a, b)$, meaning that $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$.

A related inference is *instance checking*, that is deciding whether an individual is an instance of a concept [7, 2]. Conversely, it may be necessary to find the concepts which an individual belongs to (*realization problem*), especially the most specific one:

Definition 1 (most specific concept). *Given an ABox \mathcal{A} and an individual a , the most specific concept of a w.r.t. \mathcal{A} is the concept C , denoted $MSC_{\mathcal{A}}(a)$, such that $\mathcal{A} \models C(a)$ and for any other concept D such that $\mathcal{A} \models D(a)$, it holds that $C \sqsubseteq D$.*

In some cases, the MSC may not be expressed by a finite description [2], yet it may be approximated. Generally approximations up to a certain depth k are considered, denoted MSC^k . We will generically indicate a maximal depth approximation with MSC^* .

Another inference is *retrieval* which consists in finding the extension of a given concept C , namely, all individuals a such that $\mathcal{K} \models C(a)$.

Many semantically equivalent (yet syntactically different) descriptions can be given for the same concept. Nevertheless, equivalent concepts can be reduced to a normal form by means of rewriting rules that preserve their equivalence [2]. Some notations are necessary to define the \mathcal{ALC} normal form. $\text{prim}(C)$ is the set of all the primitive concepts (and their negations) occurring at the top-level of C ; $\text{val}_R(C) = C_1 \sqcap \dots \sqcap C_n$ if there exists a value restriction $\forall R.(C_1 \sqcap \dots \sqcap C_n)$ on the top-level of C , otherwise $\text{val}_R(C) = \top$; $\text{ex}_R(C)$ is the set of the descriptions C' appearing in existential restrictions $\exists R.C'$ at the top-level conjunction of C . The normal form is defined as follows:

Definition 2 (*\mathcal{ALC} normal form*). A description C is in \mathcal{ALC} normal form iff $C \equiv \perp$ or $C \equiv \top$ or if $C = C_1 \sqcup \dots \sqcup C_n$ with

$$C_i = \prod_{P \in \text{prim}(C_i)} P \sqcap \prod_{R \in N_R} \left(\forall R. \text{val}_R(C_i) \sqcap \prod_{E \in \text{ex}_R(C_i)} \exists R.E \right)$$

where, for all $i = 1, \dots, n$, $C_i \not\equiv \perp$ and, for any $R \in N_R$, $\text{val}_R(C_i)$ and every sub-description in $\text{ex}_R(C_i)$ are in normal form.

3 Kernel Functions

In the kernel methods, the learning algorithm (inductive bias) and the choice of the kernel function (language bias) are almost completely independent. Thus, an efficient algorithm for attribute-value instance spaces can be converted into one suitable for structured spaces (e.g. trees, graphs) by merely replacing the kernel function with a suitable one. This motivates the increasing interest addressed to the SVMs and other kernel methods [3] that reproduce learning in high-dimensional spaces while working like in a vectorial representation.

Kernels are endowed with the closure property w.r.t. many operations. In particular this class is closed w.r.t. convolution [8]: such kernels can deal with compounds by decomposing them into their parts, provided that valid kernels have already been defined for them. Other works have continued this line of research introducing kernels for strings, trees, graphs and other discrete structures [4]. In particular, [5] shows how to define generic kernels based on type construction where types are defined in a declarative way. While these kernels were defined as depending on specific structures, a more flexible method is building kernels as parametrized on a uniform representation. Cumby and Roth [9] propose the syntax-driven definition of kernels based on a simple DL representation, the *Feature Description Language*. They show that the feature space blow-up is mitigated by the adoption of efficiently computable kernels. These functions transform the initial representation of the instances into the related active features, thus allowing learning the classifier directly from the structured data.

Grounded on [5], a (family of) valid kernel for the space X of \mathcal{ALC} descriptions has been proposed [6]. Recurring to the convolution kernels [8], the normal form is used to decompose complex descriptions level-wise into sub-descriptions as follows:

Definition 3 (ALC kernel). Given an interpretation \mathcal{I} , the ALC kernel based on \mathcal{I} is the function $k_{\mathcal{I}} : X \times X \mapsto \mathbb{R}$ inductively defined as follows. Let two descriptions in normal form be $D_1 = \bigsqcup_{i=1}^n C_i^1$ and $D_2 = \bigsqcup_{j=1}^m C_j^2$, then:

disjunctive descriptions: $k_{\mathcal{I}}(D_1, D_2) = \lambda \sum_{i=1}^n \sum_{j=1}^m k_{\mathcal{I}}(C_i^1, C_j^2)$ with $\lambda \in]0, 1]$

conjunctive descriptions:

$$k_{\mathcal{I}}(C^1, C^2) = \prod_{\substack{P_1 \in \text{prim}(C^1) \\ P_2 \in \text{prim}(C^2)}} k_{\mathcal{I}}(P_1, P_2) \cdot \prod_{R \in N_R} k_{\mathcal{I}}(\text{val}_R(C^1), \text{val}_R(C^2)) \cdot \prod_{R \in N_R} \sum_{\substack{C_i^1 \in \text{ex}_R(C^1) \\ C_j^2 \in \text{ex}_R(C^2)}} k_{\mathcal{I}}(C_i^1, C_j^2)$$

primitive concepts: $k_{\mathcal{I}}(P_1, P_2) = k_{\text{set}}(P_1^{\mathcal{I}}, P_2^{\mathcal{I}}) = |P_1^{\mathcal{I}} \cap P_2^{\mathcal{I}}|$

where k_{set} is the kernel for set structures defined in [5]. This case includes also the negation of primitive concepts using: $(-P)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus P^{\mathcal{I}}$

This kernel computes the similarity between disjunctive as the sum of the cross-similarities between any couple of disjuncts from either description (λ is employed to downweight the similarity of the sub-descriptions on the grounds of the level where they occur). The conjunctive kernel computes the similarity between two input descriptions, distinguishing among primitive concepts, those referred in the value restrictions and those referred in the existential restrictions. These similarity values are multiplied reflecting the fact that all the restrictions have to be satisfied at a conjunctive level. The similarity between primitive concepts is measured in terms of the intersection of their extension.

The kernel can be extended to the case of individuals $a, b \in \text{Ind}(\mathcal{A})$ simply by taking into account the approximations of their MSCs: $k_{\mathcal{I}}(a, b) = k_{\mathcal{I}}(\text{MSC}^*(a), \text{MSC}^*(b))$.

The application of the kernel function to most expressive DL is not trivial. DLs allowing normal form concept definitions can only be considered. Moreover, for each constructor not included in the ALC logic, a kernel definition has to be provided.

4 Concept Retrieval by means of Kernel Methods

SVMs are classifiers, that, exploiting a kernel function, map the training data into a higher dimensional feature space where they can be classified using a linear classifier. The SVM, as any other kernel method, can be applied to whatever knowledge representation, provided a kernel function suitable for the chosen representation. Hence, a SVM can be applied to an ALC knowledge base, considering the kernel function in Def. 3. In this paper, the SVM is used to solve the following classification problem:

Definition 4 (Problem Definition). Given a knowledge base $KB = (\mathcal{T}, \mathcal{A})$, let $\text{Ind}(\mathcal{A})$ be the set of all individuals in \mathcal{A} and $C = \{C_1, \dots, C_s\}$ the set of all concepts (both primitive and defined) in \mathcal{T} . The problem to solve is: considered an individual $a \in \text{Ind}(\mathcal{A})$ determine the set of concepts $\{C_1, \dots, C_t\} \subseteq C$ to which a belongs to.

In the general setting of SVMs, the classes for the classification are disjoint. This is not generally verified in the SW context, where an individual can be instance of more than one concept. To solve this problem, a new answering procedure is proposed. It is based on the decomposition of the multi-class problem into smaller binary classification problems (one per class). Therefore, a simple binary value set ($V = \{-1, +1\}$) can be employed, where (+1) indicates that an example x_i occurs in the ABox w.r.t. the considered concept C_j (namely $C_j(x_i) \in \mathcal{A}$); (-1) indicates the absence of the assertion in the ABox. As an alternative, it can be considered +1 when $C_j(x_i)$ can be inferred from the knowledge base, and -1 otherwise.

Another issue has to be considered. In the general classification setting an implicit assumption of *Closed World* is made. On the contrary, in the SW context the *Open World Assumption* (OWA) is generally made. To deal with the OWA, the absence of information on whether a certain instance x_i belongs to the extension of concept C_j should not be interpreted negatively, as seen before, rather, it should count as neutral information. Thus, another value set has to be considered, namely $V = \{+1, -1, 0\}$, where the three values denote, respectively, assertion occurrence ($C_j(x_i) \in \mathcal{A}$), occurrence of the opposite assertion ($\neg C_j(x) \in \mathcal{A}$) and assertion absence in \mathcal{A} . Occurrences can be easily computed with a lookup in the ABox. Moreover, as in the previous case, a more complex procedure may be devised by substituting the notion of occurrence (absence) of assertions in (from) the ABox with the one of derivability from the whole KB, i.e. $\mathcal{K} \vdash C_j(x_i)$ ($\mathcal{K} \not\vdash C_j(x_i)$), $\mathcal{K} \not\vdash C_j(x_i)$ and $\mathcal{K} \not\vdash \neg C_j(x_i)$, respectively.

Hence, considered the query instance x_q , for every concept $C_j \in \mathcal{C}$ the classifier will return +1 if x_q is an instance of C_j , -1 if x_q is an instance of $\neg C_j$, and 0 otherwise. The classification is performed on the ground of a set of training examples from which such information can be derived.

The classification results can be used to improve concept retrieval service. By classifying the individuals in the ABox w.r.t. all concepts, concept retrieval is performed exploiting an inductive approach. As will be experimentally shown in the following, the classifier, besides of having a comparable behavior w.r.t. a standard reasoner, is also able to induce new knowledge that is not logically derivable. Moreover it can be employed for the query answering task by determining, as illustrated above, the extension of a new query concept built from concepts and roles in the considered ontology.

5 Experimental Evaluation

In order to solve the classification problem presented in the previous section and assess the validity of the \mathcal{ALC} kernel function (see Def. 3), a SVM from the LIBSVM library¹ has been considered. The instance classification has been performed on nine different ontologies represented in OWL: FAMILY and UNIVERSITY handmade ontologies, FSM, SURFACE-WATER-MODEL, NEWTESTAMENTNAMES, SCIENCE, PEOPLE, NEWSPAPER and WINES ontologies from the Protégé library². Although they are represented in languages that are different from \mathcal{ALC} , constructors that are not allowed by \mathcal{ALC} are simply discarded, in order to apply the kernel function.

¹ <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

² See the webpage: <http://protege.stanford.edu/plugins/owl/owl-library>

The classification method was applied to all the individuals in each ontology; namely, the individuals were checked to assess if they were instances of the concepts in the ontology through the SVM. The performance was evaluated comparing its responses to those returned by a standard reasoner³ used as baseline. Specifically, for each individual in the ontology the MSC is computed and enlisted in the set of training (or test) examples. Each example is classified applying the SVM and the \mathcal{ALC} kernel function with $\lambda = 1$ (see Def. 3). The experiment has been repeated twice, adopting the leave-one-out cross validation procedure for ontologies with less than 50 individuals, and the ten-fold cross validation procedure for the other ontologies. For each concept in the ontology, the following parameters have been measured for the evaluation: *match rate* computed as the number of cases of individuals that got exactly the same classification by both classifiers with respect to the overall number of individuals; *omission error rate* computed as the amount of unlabeled individuals (namely the method could not determine whether it was an instance or not) while it was to be classified as an instance of that concept; *commission error rate* computed as the amount of individuals (analogically) labeled as instances of a concept, while they (logically) belong to that concept or vice-versa; *induction rate* computed as the amount of individuals that were found to belong to a concept or its negation, while this information is not logically derivable from the knowledge base. The average rates obtained over all the concepts in each ontology are reported, jointly with their range. By looking at Tab. 1, reporting the experimental outcomes, it is important to note that, for every ontology, the commission error is quite low. This means that the classifier did not make critical mistakes, i.e. cases when an individual is deemed as an instance of a concept while it really is an instance of another disjoint concept. Particularly, the commission error rate is not null in case of UNIVERSITY and FSM ontologies and consequently also the match rate is the lowest. It is worthwhile to note that these ontologies have the lowest number of individuals for concepts. Specifically, the number of concepts is almost similar to the number of individuals, this may represent a situation in which there is not enough information for separating the feature space and then produce a correct classification. However, also in this condition, the commission error is quite low, the matching rate is considerably high and the classifier is able to induce new knowledge (induction rate not null).

In general, looking at Tab. 1 it is possible to note that the match rate increases with the increase of the number of individuals in the considered ontology with a consequent strong decrease of the commission error rate that is close to 0 in such cases. Almost always the classifier is able to induce new knowledge. Anyway it presents also a conservative behavior, indeed the omission error rate is very often not null. To decrease the tendency to a conservative behavior of the classifier, a threshold could be introduced for the consideration of the "unknown" (namely labeled with 0) training examples.

Another experiment has been done, to test the method as a means for performing inductive concept retrieval w.r.t. new query concepts built from a considered ontology. The method has been applied to perform a number of retrieval problems applied to the considered ontologies using $\lambda = 1$ for the kernel function. The experiment was quite intensive involving the classification of all the individuals in each ontology; namely, the individuals were checked through the inductive procedure to assess whether they were

³ PELLET: <http://pellet.owdl.com>

Table 1. Results (average and range) of the experiments with $\lambda = 1$.

ONTOLOGY		<i>match rate</i>	<i>induction rate</i>	<i>omis. err. rate</i>	<i>comm. err. rate</i>
PEOPLE	<i>avg.</i>	0.866	0.054	0.08	0.00
	<i>range</i>	0.66 - 0.99	0.00 - 0.32	0.00 - 0.22	0.00 - 0.03
UNIVERSITY	<i>avg.</i>	0.789	0.114	0.018	0.079
	<i>range</i>	0.63 - 1.00	0.00 - 0.21	0.00 - 0.21	0.00 - 0.26
FSM	<i>avg.</i>	0.917	0.007	0.00	0.076
	<i>range</i>	0.70 - 1.00	0.00 - 0.10	0.00 - 0.00	0.00 - 0.30
FAMILY	<i>avg.</i>	0.619	0.032	0.349	0.00
	<i>range</i>	0.39 - 0.89	0.00 - 0.41	0.00 - 0.62	0.00 - 0.00
NEWSPAPER	<i>avg.</i>	0.903	0.00	0.097	0.00
	<i>range</i>	0.74 - 0.99	0.00 - 0.00	0.02 - 0.26	0.00 - 0.00
WINES	<i>avg.</i>	0.956	0.004	0.04	0.00
	<i>range</i>	0.65 - 1.00	0.00 - 0.27	0.01 - 0.34	0.00 - 0.00
SCIENCE	<i>avg.</i>	0.942	0.007	0.051	0.00
	<i>range</i>	0.80 - 1.00	0.00 - 0.04	0.00 - 0.20	0.00 - 0.00
S.-W.-M.	<i>avg.</i>	0.871	0.067	0.062	0.00
	<i>range</i>	0.57 - 0.98	0.00 - 0.42	0.00 - 0.40	0.00 - 0.00
N.T.N.	<i>avg.</i>	0.925	0.026	0.048	0.001
	<i>range</i>	0.66 - 0.99	0.00 - 0.32	0.00 - 0.22	0.00 - 0.03

Table 2. Results (average) of the querying experiments.

ONTOLOGY	<i>match rate</i>	<i>ind. rate</i>	<i>omis. err. rate</i>	<i>comm. err. rate</i>
PEOPLE	0.886	0.040	0.074	0.0
UNIVERSITY	0.72	0.16	0.009	0.111
FSM	0.878	0.009	0.0	0.114
FAMILY	0.663	0.045	0.292	0.0
NEWSPAPER	0.779	0.0	0.221	0.0
WINES	0.943	0.0	0.057	0.0
SCIENCE	0.978	0.005	0.016	0.0
S.-W.-M.	0.804	0.134	0.062	0.0
NTN	0.906	0.022	0.072	0.0

retrieved as instances of a query concept. Therefore, 15 queries were randomly generated by means of conjunctions/disjunctions of primitive and/or defined concepts of each ontology. As for the previous experiment, the leave-one-out procedure was performed in case of ontologies with less than 50 individuals and a ten-fold cross validation was performed for the others. The outcomes are reported in Tab. 2, from which it is possible to observe that the behavior of the classifier mainly remains the same as in the experiment whose outcomes are reported in Tab. 1.

Summarizing, the \mathcal{ALC} kernel function can be effectively used, jointly with a SVM, to perform inductive concept retrieval, guaranteeing almost null commission error and interestingly the ability to induce new knowledge. The performance of the classifier increases with the increase of the number of individuals populating the considered ontology that have to be preferable homogeneously spread w.r.t. the concept in the ontology.

6 Conclusions and Future Work

In this work we have tested a kernel function for \mathcal{ALC} descriptions integrated with a SVM in a (multi)relational learning setting. The resulting classifier has been used to improve concept retrieval and query answering tasks in the ontological setting. It has been experimentally shown that its performance is not only comparable to the one of a standard reasoner, but it is also able to induce new knowledge, which is not logically derivable. Particularly, an increase in prediction accuracy was observed when the instances are homogeneously spread.

The realized classifier can be exploited for predicting/suggesting missing information about individuals, thus completing large ontologies. Specifically, it can be used to semi-automatize the population of an ABox. Indeed, the new assertions can be suggested to the knowledge engineer that has only to validate their inclusion. This constitutes a new approach in the SW context, since the efficiency of the statistical and numerical approaches and the effectiveness of a symbolic representation have been combined.

The main weakness of the approach is on its scalability towards more complex DLs. While computing MSC approximations might be feasible, it may be more difficult focusing on a normal form when comparing descriptions. Indeed, as long as the expressivity increases, the gap between syntactic structure semantics of the descriptions becomes more evident. As a next step, we can foresee the investigation of defining kernels for more expressive languages w.r.t. \mathcal{ALC} , e.g. languages enriched with (qualified) number restrictions and inverse roles [2].

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Scientific American* **284** (2001) 34–43
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook. Cambridge University Press (2003)
3. Schölkopf, B., Smola, A.: *Learning with Kernels*. The MIT Press (2002)
4. Gärtner, T.: A survey of kernels for structured data. *SIGKDD Explorations* **5** (2003) 49–58
5. Gärtner, T., Lloyd, J., Flach, P.: Kernels and distances for structured data. *Machine Learning* **57** (2004) 205–232
6. Fanizzi, N., d’Amato, C.: A declarative kernel for \mathcal{ALC} concept descriptions. In Esposito, F., Raś, Z.W., Malerba, D., (Eds.), G.S., eds.: *In Proceedings of the 16th International Symposium on Methodologies for Intelligent Systems*. Volume 4203 of *Lecture Notes in Computer Science*, Springer (2006) 322–331
7. Donini, F.M., Lenzerini, M., Nardi, D., Schaerf, A.: Deduction in concept languages: From subsumption to instance checking. *Journal of Logic and Computation* **4** (1994) 423–452
8. Haussler, D.: Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, Department of Computer Science, University of California – Santa Cruz (1999)
9. Cumby, C., Roth, D.: On kernel methods for relational learning. In Fawcett, T., N.Mishra, eds.: *Proceedings of the 20th International Conference on Machine Learning, ICML2003*, AAAI Press (2003) 107–114